



PENDETEKSIAN GAWANG MENGGUNAKAN ALGORITMA *RANSAC* PADA *PLATFORM* *DARWIN-OP* BERBASIS PERATURAN KRSBI 2016

PENYUSUN :

Uti Solichah

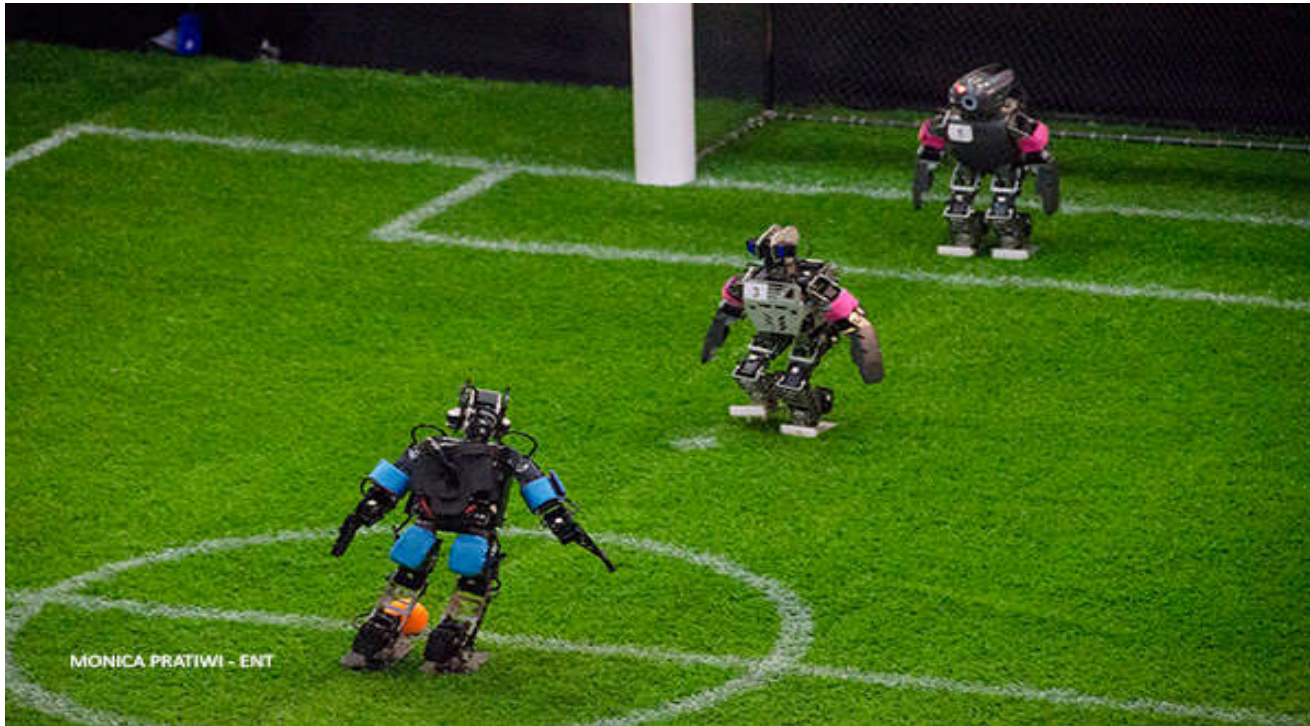
PEMBIMBING :

Dr. Eng. Nanik Suciati S.Kom., M.Kom

Muhtadin., S.T., M.T



LATAR BELAKANG



Gambar 1. Robot Ichiro dalam Kontes Robot Sepak Bola Indonesia 2016

Robot dituntut bergerak secara *autonomous*



KEMAMPUAN DASAR ROBOT

BOLA	GARIS LAPANGAN	GAWANG
------	----------------	--------

Permasalahan

Metode *Hough Transform* yang sudah ada belum optimal. Terutama pada :

1. Ketepatan jarak gawang dengan robot
2. Performa kecepatan pendeteksian gawang

Diperlukan pendekatan metode lain untuk mendeteksi gawang



*“Pendeteksian Gawang Menggunakan Algoritma
Ransac pada Platform Darwin-OP Berbasis
Peraturan KRSBI 2016”*

TUJUAN

*“Mengaplikasikan Algoritma Ransac pada pendeteksian
Pada platform Robot Darwin-OP”*

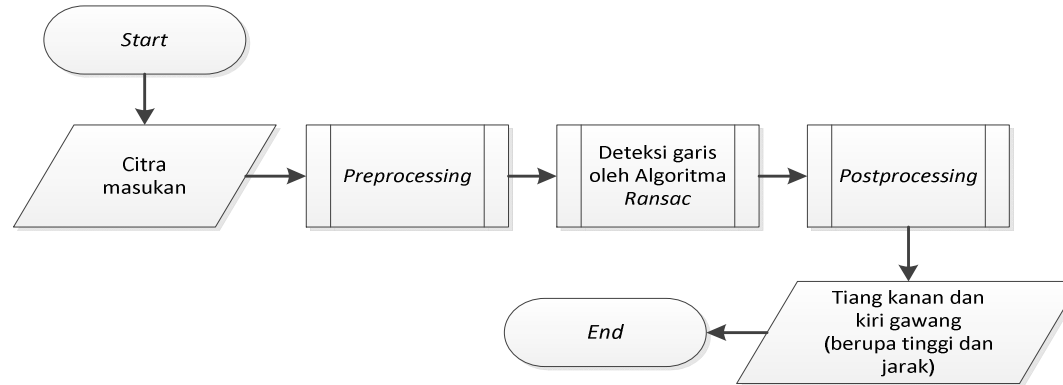


MANFAAT

*“Menemukan metode alternatif pendeteksian gawang
Pada platform Robot Darwin-OP”*

*Mempercepat dan mempermudah self localization Robot
Meningkatkan performa bermain Tim Robot Sepak Bola ITS*

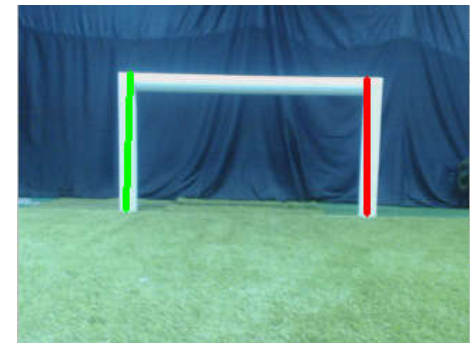
DESAIN **SISTEM**



Gambar 2. Diagram alur desain sistem deteksi gawang



Gambar 3. Contoh citra masukan gawang



Gambar 4. Contoh citra keluaran gawang



1 *PREPROCESSING*



PREPROCESSING

Grayscale citra

Thresholding Citra

Reduksi *Noise* Citra

Deteksi Tepi Citra

Transformasi Citra ke *point*

Grayscale Citra

Pengganti proses segmentasi warna dalam *colorspace hsv* pada robot

Membantu memudahkan dalam proses *thresholding*



Gambar 5. Citra masukan



Gambar 6. Citra hasil *thresholding*



PREPROCESSING

Grayscale citra

Thresholding Citra

Reduksi *Noise* Citra

Deteksi Tepi Citra

Transformasi Citra ke *point*

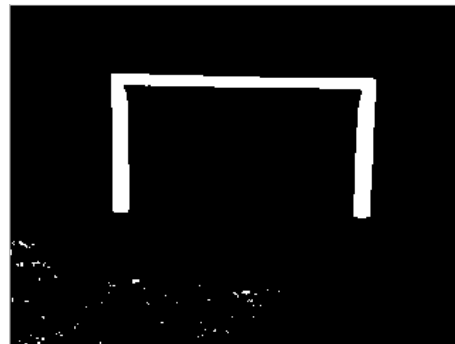
Thresholding Citra

Proses segmentasi warna gawang

Membantu memisahkan objek gawang dengan selain nya



Gambar 7. Citra hasil *grayscale*



Gambar 8. Citra hasil *thresholding*



PREPROCESSING

Grayscale citra

Thresholding Citra

Reduksi Noise Citra

Deteksi Tepi Citra

Transformasi Citra ke point

Reduksi Noise

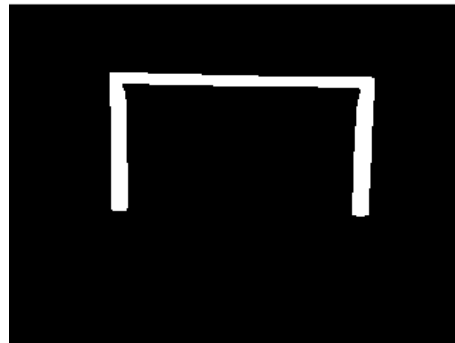
Proses penghilangan *noise* pada citra

Membantu sistem mendapatkan hasil segmentasi yang ideal

Meminimalkan kemungkinan *error* deteksi garis akibat *noise*



Gambar 9. Citra sebelum reduksi *noise*



Gambar 10. Citra setelah reduksi *noise*

PREPROCESSING

Grayscale citra

Thresholding Citra

Reduksi Noise Citra

Deteksi Tepi Citra

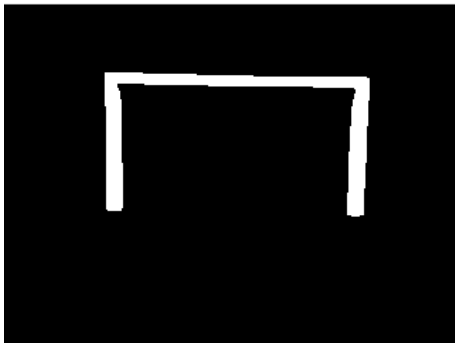
*Transformasi Citra ke **point***

Deteksi Tepi Citra

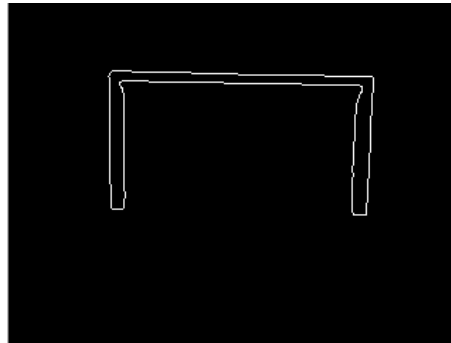
Proses pendeteksian tepi pada citra

Menggunakan algoritma *canny* untuk deteksi tepi

Meminimalkan jumlah *point* yang akan di proses pada deteksi garis



Gambar 11. Citra sebelum deteksi tepi



Gambar 12. Citra setelah deteksi tepi



PREPROCESSING

Grayscale citra

Thresholding Citra

Reduksi *Noise* Citra

Deteksi Tepi Citra

Transformasi Citra ke *point*

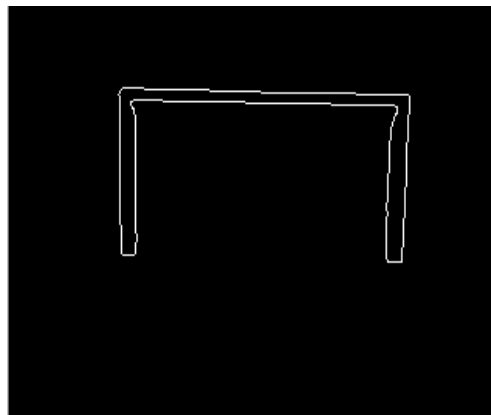
Transformasi citra ke point

Mentransformasikan citra gambar ke bentuk himpunan *point* (x,y)

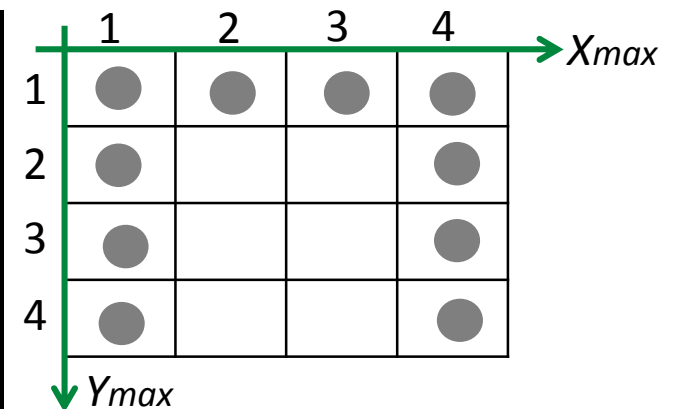
Memudahkan untuk pemrosesan deteksi garis oleh Algoritma *Ransac*

Tabel 1. Cuplikan himpunan
point Gambar 6

Point	x	y
1	70	51
2	71	49
3	71	50
4	71	52
5	71	53
6	71	54
7	71	55
8	71	56
9	71	57
10	71	58
11	71	59
12	71	60



Gambar 13. Citra hasil deteksi tepi

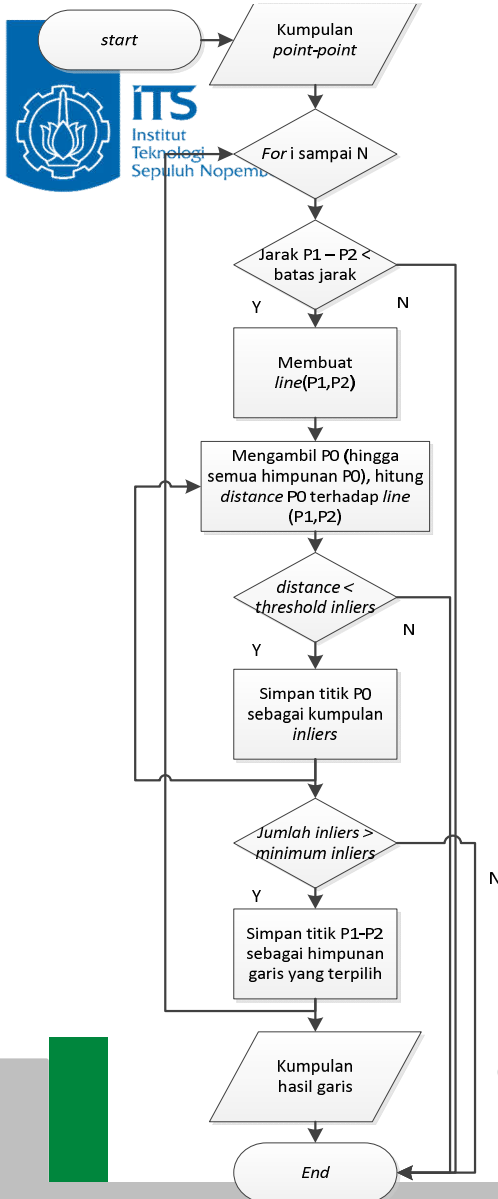


Gambar 14. Ilustrasi transformasi citra ke *point*

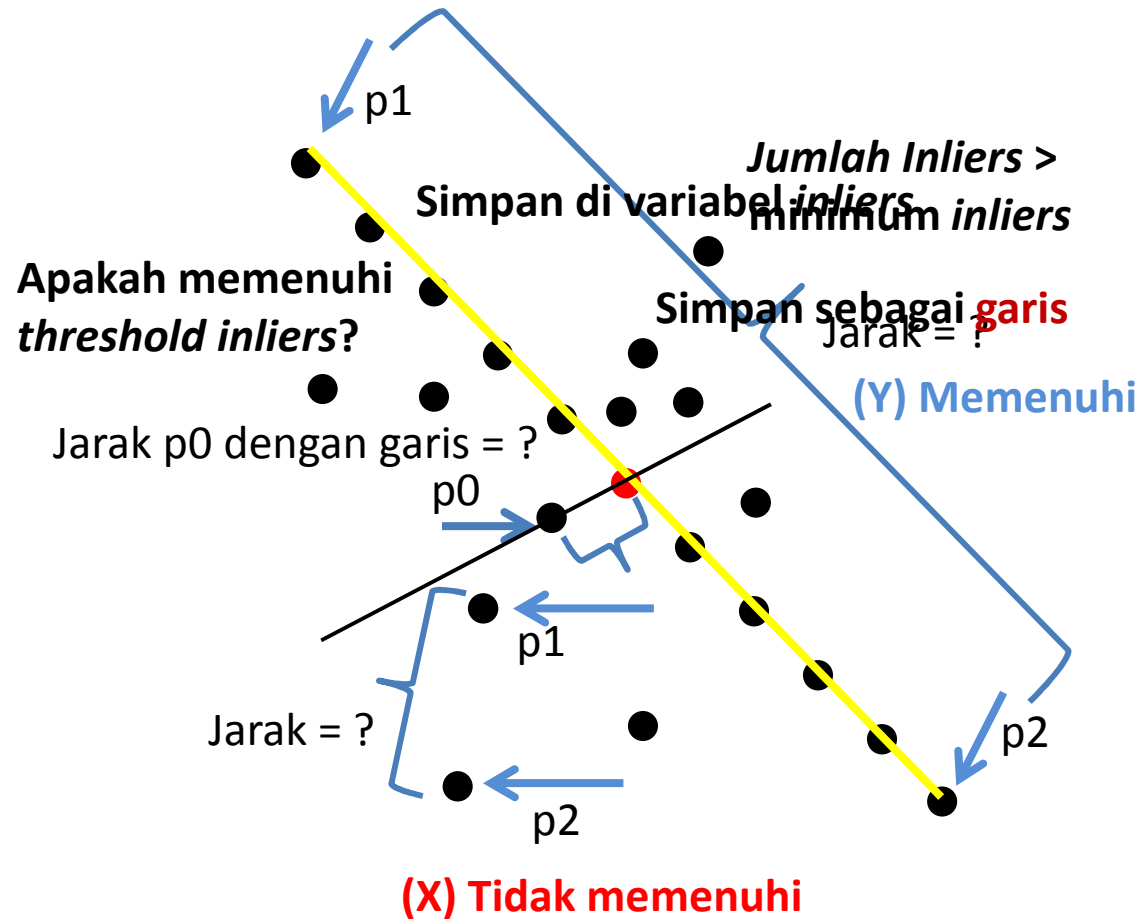


2DETEKSI GARIS

MENGGUNAKAN ALGORITMA RANSAC



Gambar 15. Diagram alur deteksi garis menggunakan Algoritma Ransac



Gambar 16. Ilustrasi deteksi garis menggunakan Algoritma Ransac



3 *POSTPROCESSING*





POSTPROCESSING

Clusterring Line

Mencari Mean *Cluster*

Drawing Line

Hitung Jarak Tiang

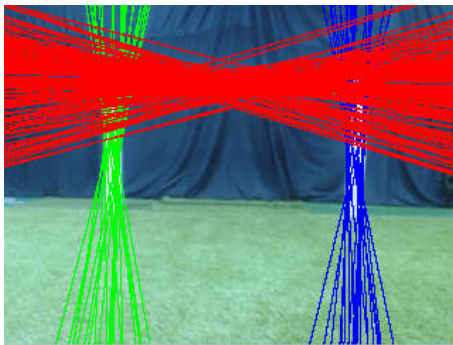
Hitung Tinggi Tiang

Clustering Line

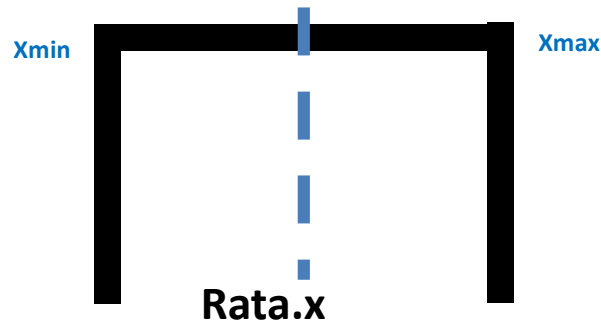
Proses menemukan garis terbaik tiang kanan dan tiang kiri

Didapat dari seleksi Rata.x gawang. $\text{Rata.x} = (\text{Xmin} + (\text{Xmax} - \text{Xmin})/2)$

Apabila $p1.x$ dan $p2.x > \text{rata.x}$ maka termasuk tiang kanan
Apabila $p1.x$ dan $p2.x < \text{rata.x}$ maka termasuk tiang kiri
Sisanya, termasuk tiang atas



Gambar 17. Citra hasil *clustering line*



Gambar 18. Ilustrasi *clustering line*

POSTPROCESSING

Clusterring Line

Mencari Mean *Cluster*

Drawing Line

Hitung Jarak Tiang

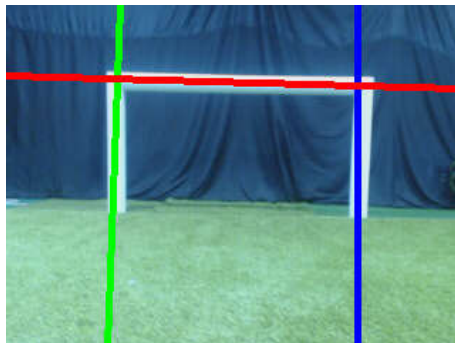
Hitung Tinggi Tiang

Mencari Mean Cluster

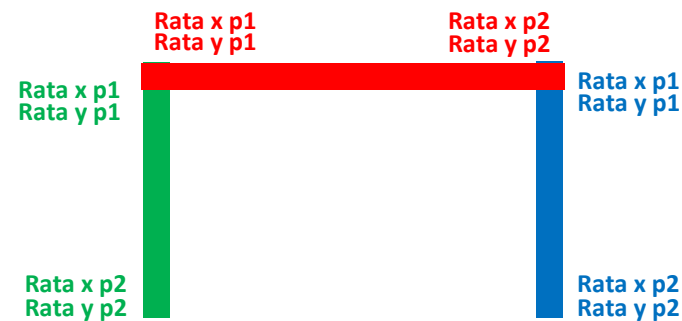
Proses Mencari mean *cluster* tiap tiang kanan dan tiang kiri

Didapat dari rata-rata x dari p1 dan rata-rata y dari p1 untuk tiap *cluster*

Didapat dari rata-rata x dari p2 dan rata-rata y dari p2 untuk tiap *cluster*



Gambar 19. Citra hasil mean *cluster*



Gambar 20. Ilustrasi mencari mean *cluster*



POSTPROCESSING

Clusterring Line

Mencari Mean *Cluster*

Drawing Line

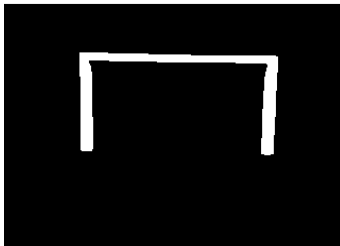
Hitung Jarak Tiang

Hitung Tinggi Tiang

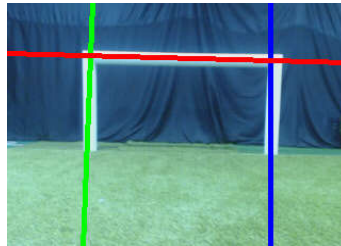
Drawing Line

Menggambar dari hasil garis mean tiang

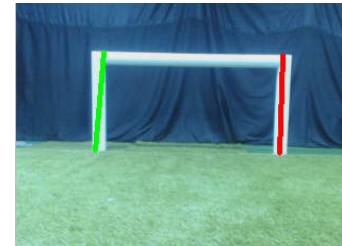
Mengecek nilai $rgb = 255$ dari hasil *thresholding* untuk mendapatkan Mulai kanan, akhir kanan, mulai kiri, akhir kiri.



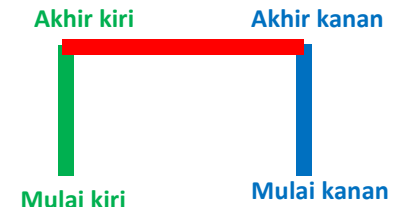
Gambar 23. Citra hasil *thresholding*



Gambar 24. Citra hasil mean cluster



Gambar 25. Hasil *drawing line*



Gambar 26. Ilustrasi *drawing line*

POSTPROCESSING

Clusterring Line

Mencari Mean *Cluster*

Drawing Line

Hitung Jarak Tiang

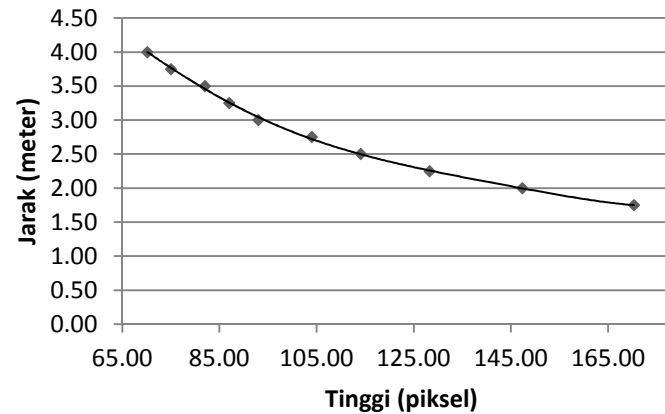
Hitung Tinggi Tiang

Hitung Jarak Tiang

Didapat dari Rumus Regresi Polynomial

Tabel 2. hubungan tinggi tiang dan jarak

No	Tinggi (piksel)	Jarak sebenarnya (meter)
1	170,35	1,75
2	147,34	2,00
3	128,25	2,25
4	114,07	2,50
5	104,02	2,75
6	93,05	3,00
7	87,01	3,25
8	82,02	3,50
9	75,06	3,75
10	70,18	4,00



Gambar 22. hasil regresi polynomial orde 5

$$\text{Jarak} = (5,04 \times 10^{-10})X^5 - (2,94 \times 10^{-7})X^4 + (6,53 \times 10^{-5})X^3 - (6,60 \times 10^{-3})X^2 + 0,25948X + 2,0383$$

POSTPROCESSING

Clusterring Line

Mencari Mean *Cluster*

Drawing Line

Hitung Jarak Tiang

Hitung Tinggi Tiang

Hitung Tinggi Tiang

Proses menghitung tinggi tiang

Dilihat dari citra *thresholding* dan mengambil batas bawah dan atas tiap tiang

Tinggi = jarak antara dua titik batas atas dan batas bawah

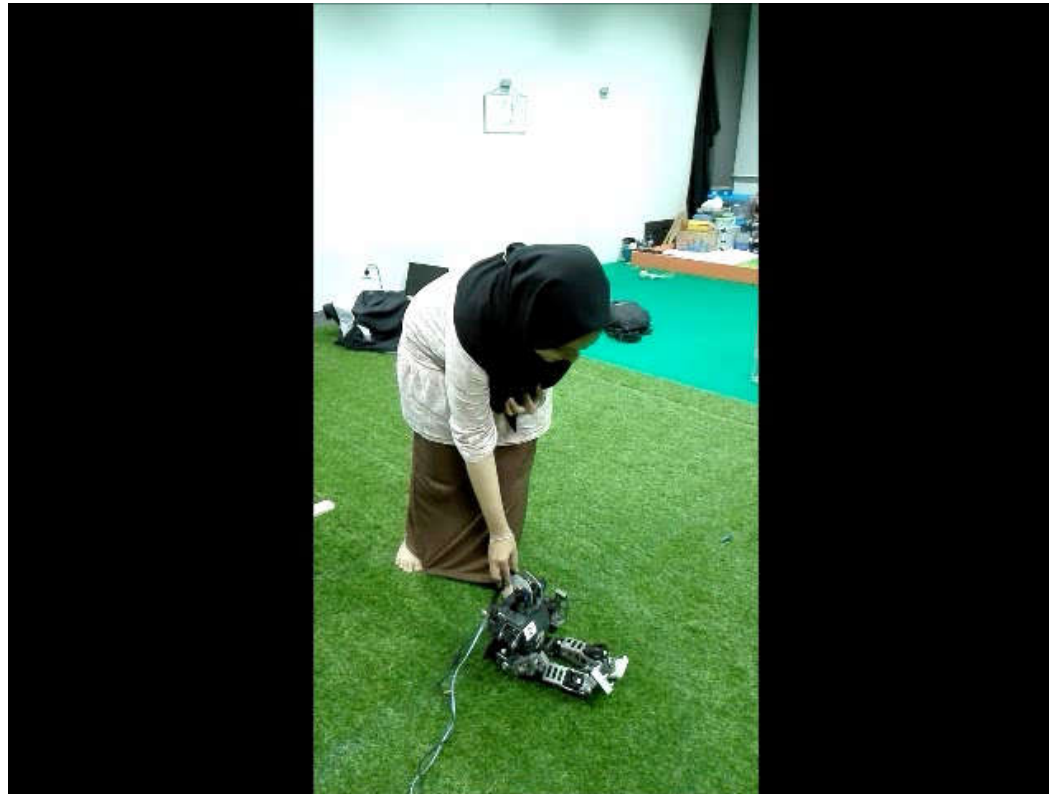
$$\text{Tinggi } T.\text{kanan} = \sqrt{(\text{mulai kanan}.x - \text{akhir kanan}.x)^2 + (\text{mulai kanan}.y - \text{akhir kanan}.y)^2}$$



Gambar 21. Ilustrasi tinggi tiang

video

Video Pendeteksian Gawang Menggunakan Algoritma *Ransac* pada *platform Darwin-OP*



PENGUJIAN



HASIL UJI COBA PARAMETER

THRESHOLDING

Parameter *thresholding* akan berubah mengikuti intensitas inputan citra

Tabel 3. Parameter *thresholding* citra

No	Nama Gambar	Batas Bawah	Batas Atas	Keterangan
1	<i>Image-1</i>	20	255	Terlalu banyak <i>noise</i> pada citra
2	<i>Image-1</i>	40	255	Banyak <i>noise</i> pada citra
3	<i>Image-1</i>	60	255	Banyak <i>noise</i> pada citra
4	<i>Image-1</i>	80	255	Banyak <i>noise</i> pada citra
7	<i>Image-1</i>	100	255	Banyak <i>noise</i> pada citra
8	<i>Image-1</i>	120	255	Masih ada <i>noise</i> pada citra
9	<i>Image-1</i>	130	255	<i>Noise</i> Citra tinggal sedikit
10	<i>Image-1</i>	140	255	Tidak ada <i>noise</i>
11	<i>Image-1</i>	160	255	Citra gawang terdegradasi
12	<i>Image-1</i>	170	255	Hanya ada tiang atas gawang
13	<i>Image-1</i>	200	255	Hasil <i>thresholding</i> sedikit
12	<i>Image-1</i>	210	255	Hasil <i>thresholding</i> sedikit
13	<i>Image-1</i>	230	255	Tidak ada hasil <i>thresholding</i>

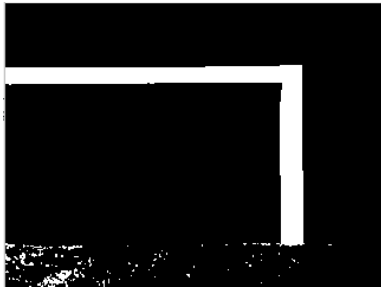


Gambar 27. Hasil *thresholding* batas bawah =100



Gambar 28. Hasil *thresholding* batas bawah = 140

REDUKSI NOISE



Gambar 29. Hasil *thresholding* batas bawah = 100

DETEKSI TEPI CITRA



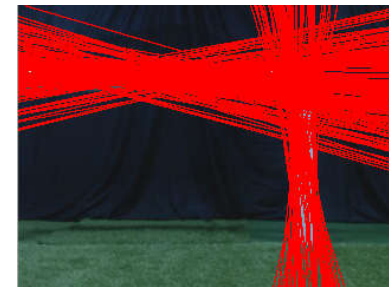
Gambar 30. Hasil *thresholding* batas bawah = 140

PARAMETER RANSAC (N)

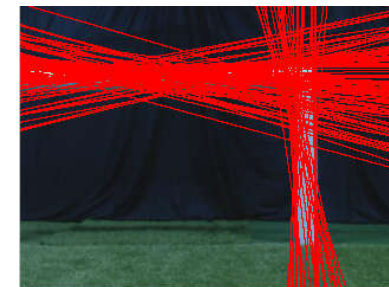
Parameter N merepresentasikan jumlah perulangan mengambil 2 titik random

Tabel 4. Parameter Perulangan minimum garis

No	Besarnya N	Keterangan
1	100	Terlalu sedikit garis yang terbuat
2	200	Garis yang terbuat pada tiang kanan dan tiang kiri terlalu sedikit
3	300	Garis yang terbuat belum ideal
4	400	Garis yang terbuat sudah ideal
5	1000	Terlalu banyak garis yang terbuat



Gambar 31. Hasil Parameter N =1000



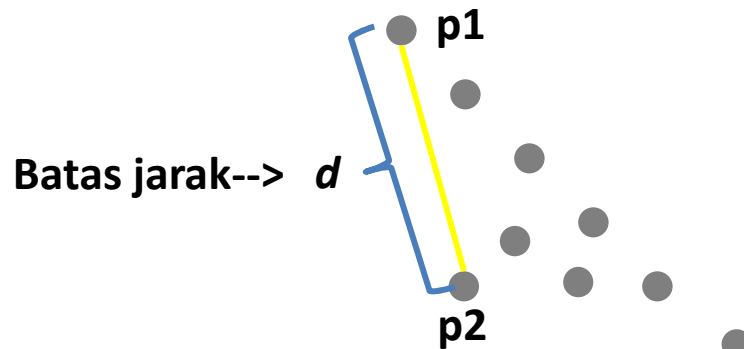
Gambar 32. Hasil Parameter N = 400

PARAMETER RANSAC (BATAS JARAK)

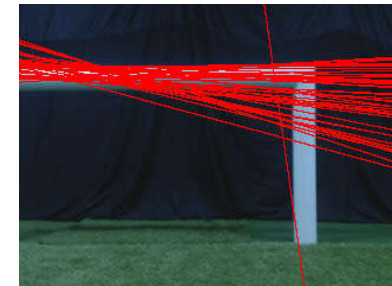
Parameter batas jarak maksimum merepresentasikan besarnya jarak piksel antara titik P1 dengan titik p2

Tabel 5. Parameter minimum batas jarak p1 dan p2

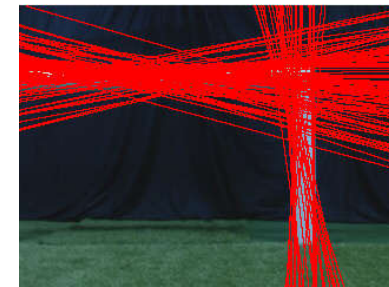
No	Batas jarak	Keterangan
1	10	Terlalu banyak garis yang dibuat
2	20	Terlalu banyak garis yang dibuat
3	40	Garis yang dibuat ideal
4	70	Sedikit garis yang dibuat
5	100	Terlalu sedikit garis yang dibuat



Gambar 33. Ilustrasi parameter batas jarak



Gambar 34. Hasil parameter batas jarak = 100



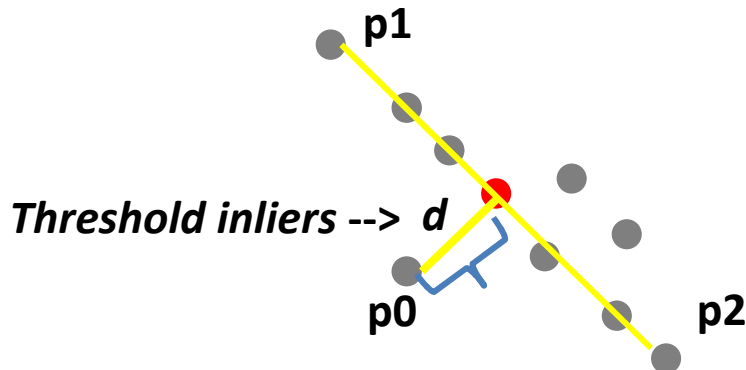
Gambar 35. Hasil parameter batas jarak = 40

PARAMETER RANSAC (THRESHOLD INLIERS)

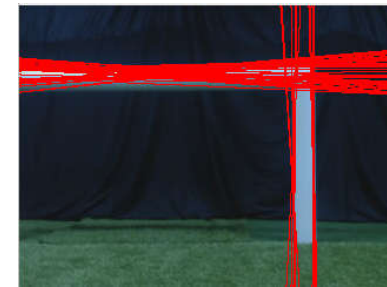
Parameter *threshold inliers* merepresentasikan jarak maksimum p_0 dengan garis p_1, p_2

Tabel 6. Parameter minimum *threshold inliers*

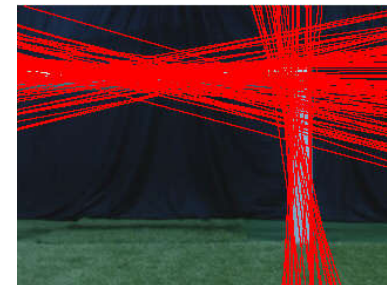
No	<i>Threshold inliers</i>	Keterangan
1	2	Sedikit garis yang terbentuk
2	5	Garis yang terbentuk ideal
3	10	Terlalu banyak garis yang terbentuk



Gambar 36. Ilustrasi parameter *threshold inliers*



Gambar 37. Hasil Parameter *threshold inliers* = 2



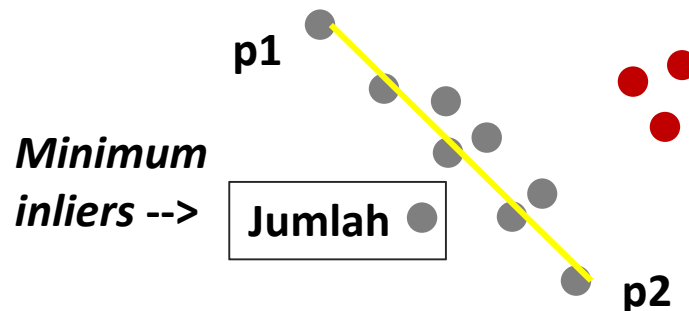
Gambar 38. Hasil Parameter *threshold inliers* = 5

PARAMETER RANSAC (MINIMUM INLIERS)

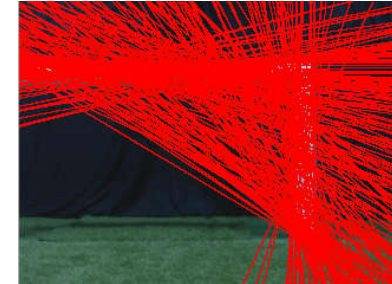
Parameter *minimum inliers* merepresentasikan jumlah titik minimum p0 untuk sebuah garis

Tabel 7. Parameter jumlah *minimum inliers*

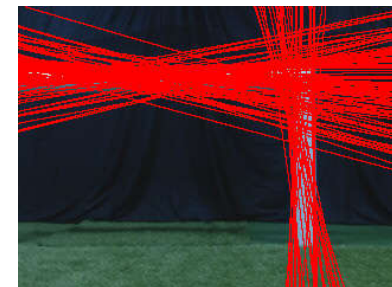
No	<i>Minimum inliers</i>	Keterangan
1	10	Terlalu banyak garis yang terbuat
2	20	Terlalu banyak garis yang terbuat
3	30	Banyak garis yang terbuat
4	40	Banyak garis yang terbuat
5	60	Cukup banyak garis yang terbuat
6	85	Garis yang terbuat ideal
7	100	Garis yang terbuat sedikit
8	200	Garis yang terbuat hanya di tiang atas saja
9	1000	Tidak ada garis yang terbuat



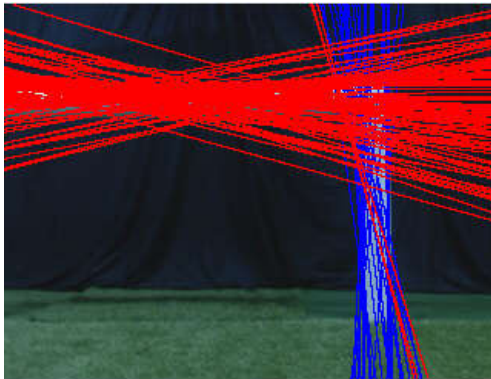
Gambar 39. ilustrasi parameter *minimum inliers*



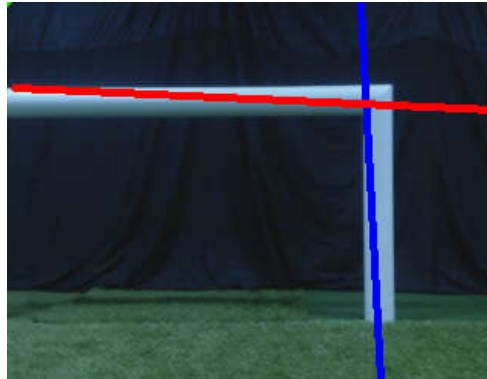
Gambar 40. Hasil Parameter *minimum inliers* = 10



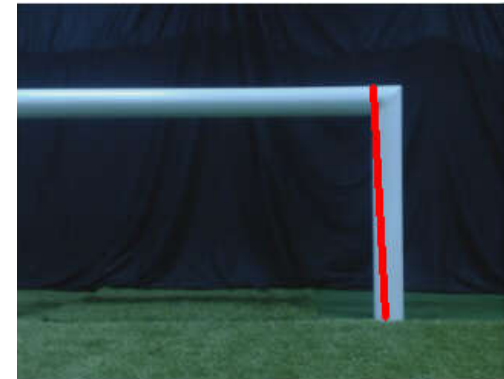
Gambar 41. Hasil Parameter *minimum inliers* = 85



Gambar 42. Hasil *clustering line*



Gambar 43. Hasil *mean tiang*



Gambar 44. Hasil *drawing line*

Tinggi yang terdeteksi = 147,217 piksel
Jarak yang terdeteksi = 1,9984 meter

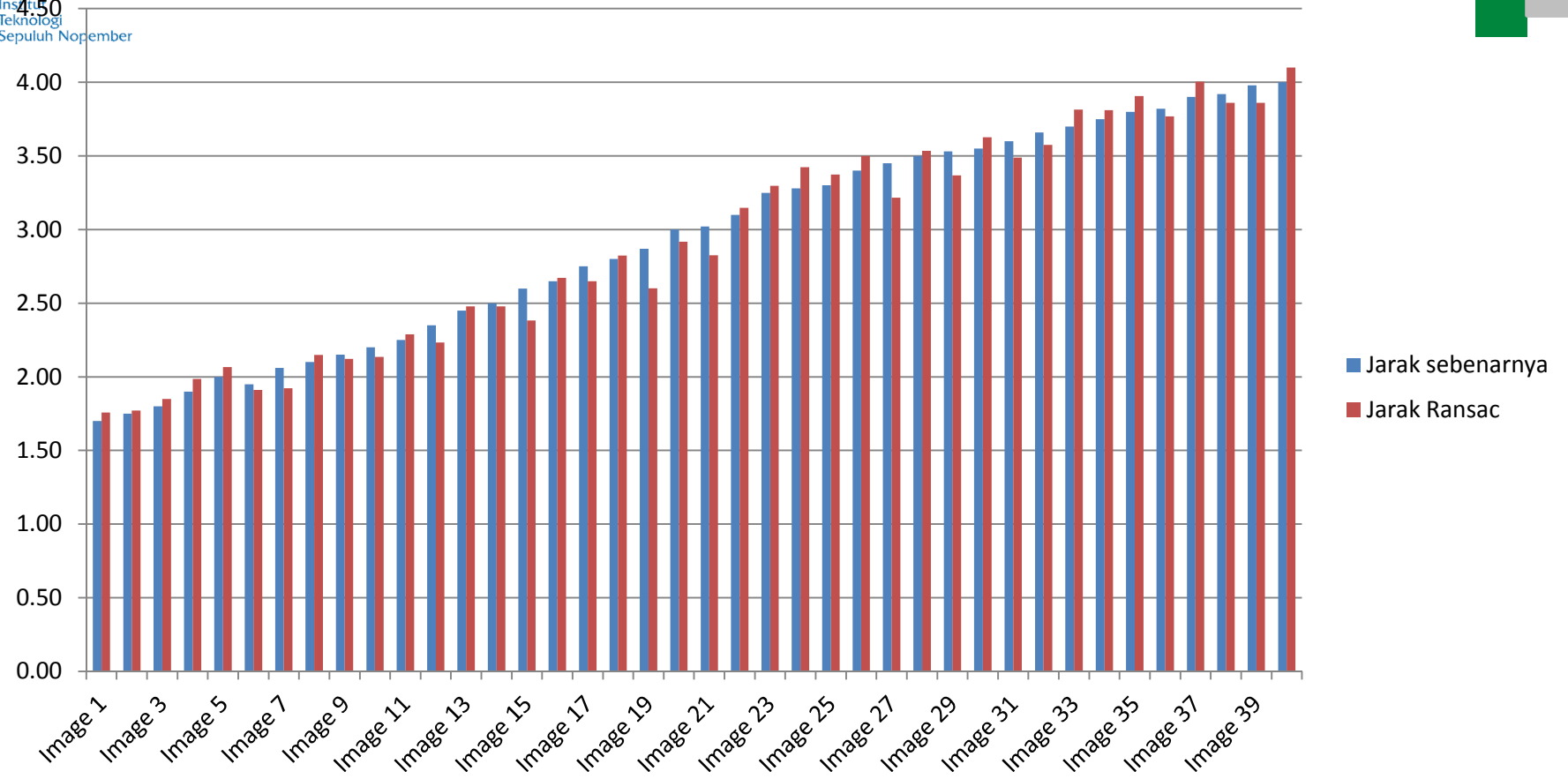


ESTIMASI JARAK



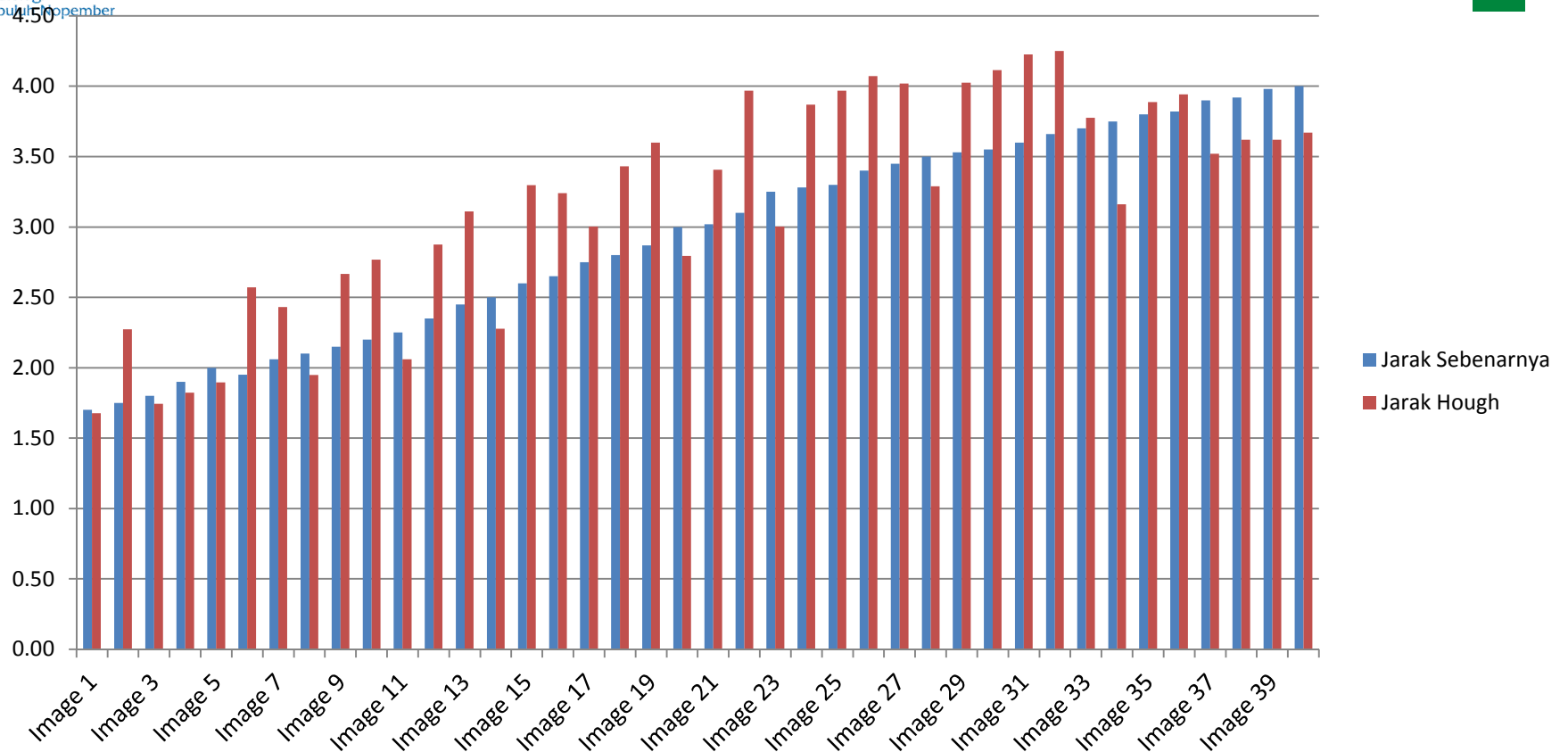
ITS

Ins4.50
Teknologi
Sepuluh Nopember



Gambar 45. Hasil uji coba jarak gawang Algoritma *Ransac*

Rata-Rata Galat *Ransac* = 0,0362 m

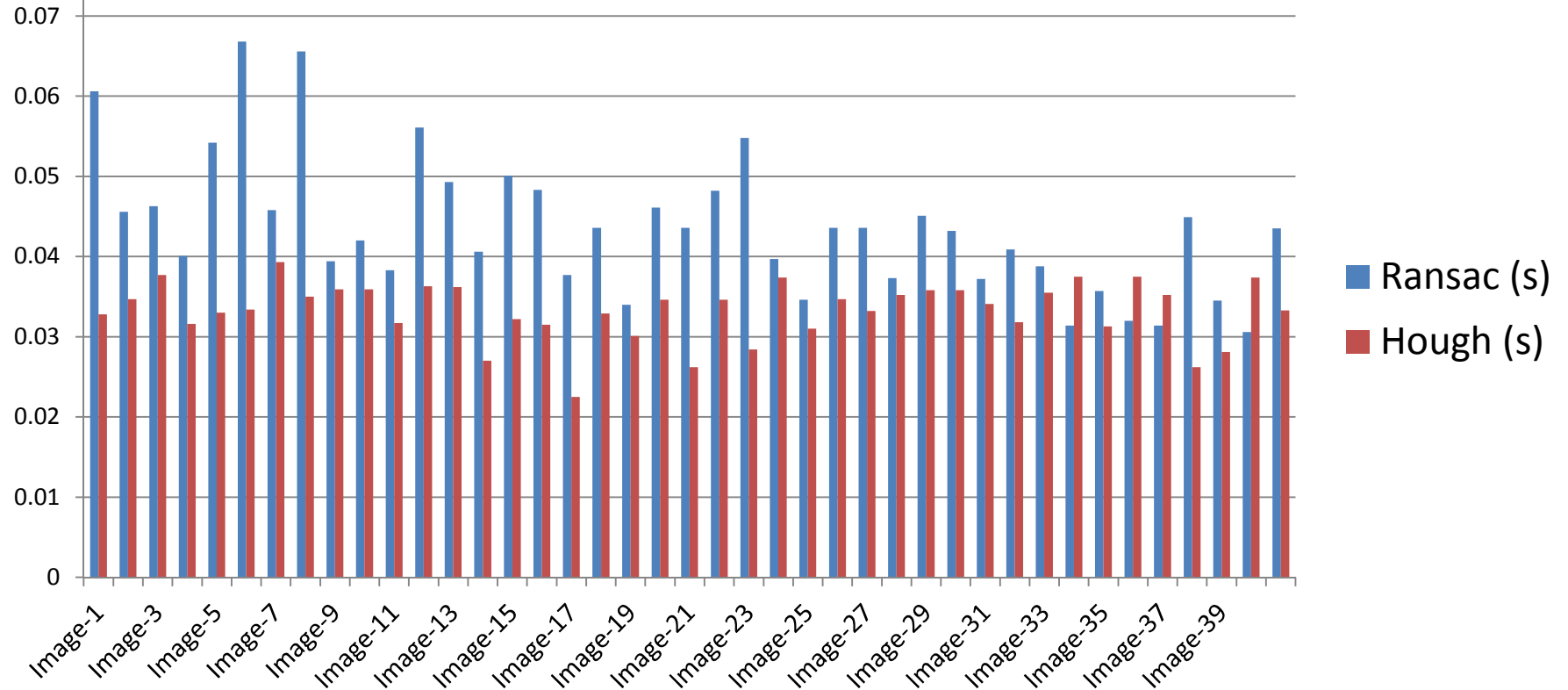


Gambar 46. Hasil uji coba jarak gawang Metode *Hough Transform*

Rata-Rata Galat *Hough Transform* = 0,1452 m



ESTIMASI WAKTU



Gambar 47. Hasil uji coba kecepatan waktu deteksi Algoritma Ransac dan Metode Hough Transform

Rata-Rata Waktu Ransac = 0,0435 | **Rata-Rata Waktu Hough Transform = 0,0333**

KESIMPULAN

Tahap *thresholding* penting, hasil parameter terbaik batas bawah *thresholding* 140, namun sangat tergantung keadaan. Hasil *thresholding* yang tidak baik akan memengaruhi hasil garis yang didapatkan.

Berdasarkan Tabel 5.3 hasil parameter N terbaik pada skenario uji coba 1 adalah 400. Representasi nilai ideal terbaik dengan hasil yang sama dan lebih cepat dibandingkan lebih dari 400.

Berdasarkan Tabel 5.4, hasil parameter batas jarak terbaik pada skenario uji coba 1 adalah 40. Nilai lebih kecil menghasilkan garis lebih banyak, dan sebaliknya. Akan memengaruhi proses mencari mean tiang nya.

Berdasarkan Tabel 5.5, hasil parameter *threshold inliers* terbaik pada skenario uji coba 1 adalah 5. Nilai lebih kecil, garis yang terbuat sedikit, dan tidak mewakili, nilai lebih besar terlalu banyak garis.

KESIMPULAN

Berdasarkan Tabel 5.6, hasil parameter minimum inliers terbaik pada skenario uji coba 1 adalah 85. nilai lebih kecil menghasilkan garis yang tidak beraturan

Berdasarkan Tabel 5.7 pada skenario uji coba 2. Algoritma Ransac memiliki keakuratan jarak robot lebih baik daripada Metode *Hough Transform*. Rata-rata galat jarak 0,03062 untuk Algoritma Ransac dan rata-rata galat jarak sebesar 0,1452 untuk Metode *Hough Transform*.

Berdasarkan Tabel 5.8 pada skenario uji coba 2. *Hough Transform* lebih cepat dibandingkan dengan metode Algoritma *Ransac*, dengan rata-rata waktu eksekusi program 0,0333 untuk Metode *Hough Transform* dan rata-rata waktu eksekusi program 0,0435 untuk Algoritma *Ransac*.

SARAN

Memperbaiki proses segmentasi warna gawang untuk mendapatkan citra ideal sebelum proses Algoritma *Ransac*

Menemukan filter yang baik dalam mendeteksi warna putih gawang pada tahap *preprocessing*

Penerapan pendeteksian gawang pada *platform* Robot *Darwin-OP* dalam permainan sepak bola robot.

CAPAIAN

Sudah membandingkan performa ketepatan jarak dan kecepatan waktu antara Algoritma *Ransac* dengan Metode *Hough Transform*

Algoritma *Ransac* sudah diimplementasikan pada *platform* Robot *Darwin-OP*

Judul ini sudah dipresentasikan pada ISRSC (*The 4th Indonesian Symposium on Robot Soccer Competition 2016*)



TERIMA KASIH

